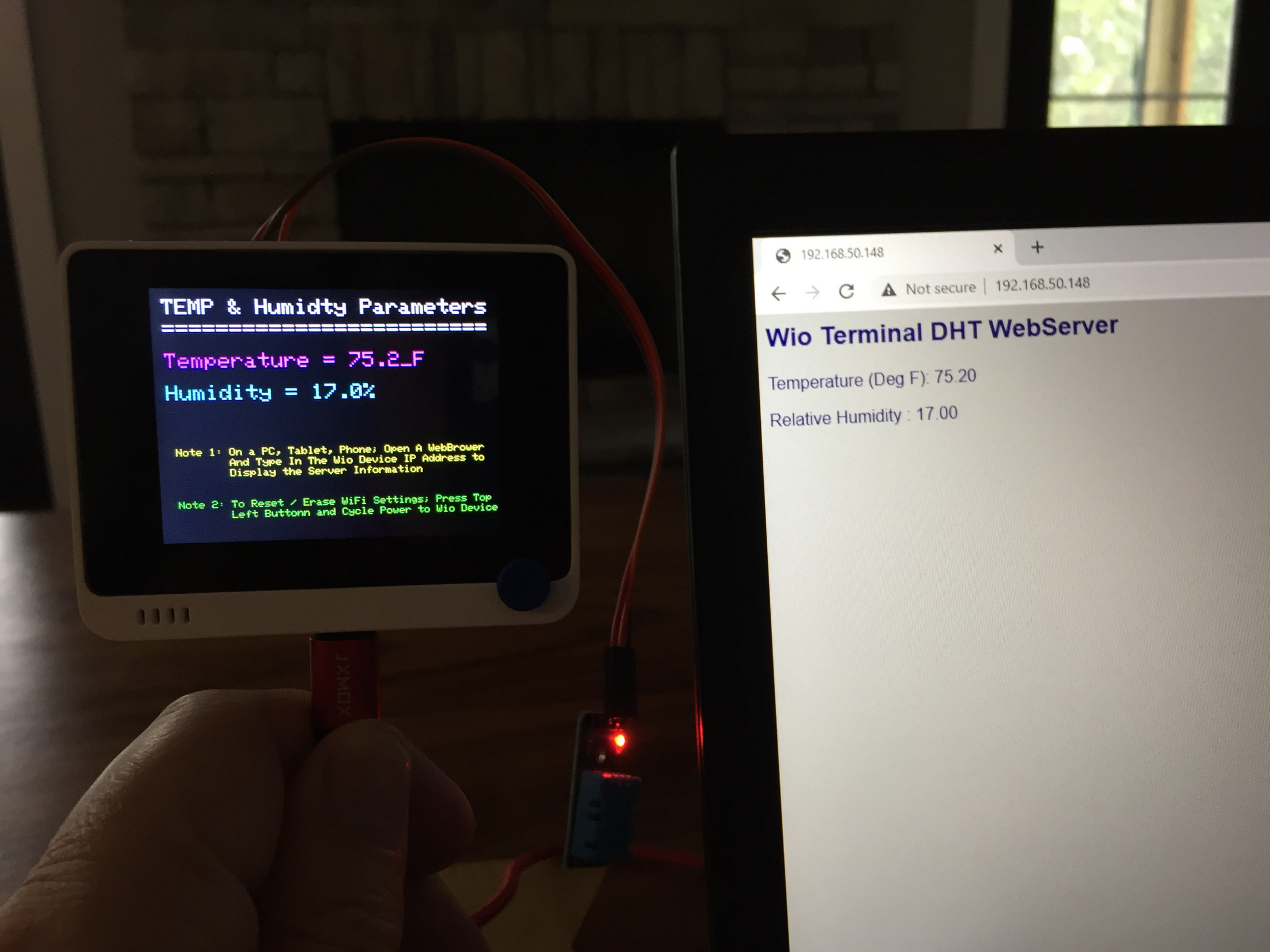
TEMP & Humidty Parameters
Temperature = 75.2_F
Humidity = 17.0%
Note 1: On a PC, Tablet, Phone; Open A WebBrower And Type In The Wio Device IP Address to Display the Server Information
Note 2: To Reset / Erase WiFi Settings; Press Top Left Buttonn and Cycle Power to Wio Device
192.168.50.148
Not secure | 192.168.50.148
Wio Terminal DHT WebServer
Temperature (Deg F): 75.20
Relative Humidity : 17.00

```
/*
 * This is a program that monitors a DHT sensor for Temperature and Humidity
 * and works with many different sensors (just uncomment out sensor type and
 * change the data pin number if not using the same one "D0")
 *
 * First Setup Wio Terminal to work with Arduino Compiler via Web Instructions
 *
 * The Program Offers the following Functions:
 * -------------------------------------------
 * 1) Displays the Temperature and Humidty Locally on Display
 * 2) Contains on-Screen Instructions for setting Up WiFi.
 * 3) Diplays common error messages for Server and Wifi.
 * 4) Detects Sensor failure and displays an error message.
 * 5) Top Left Buttun Resets WiFi Settings.
 * 6) The Web-Server operates by typing in the Wio Device IP
 *    Address in a Browser Window and Displays the current
 *    Temperature and Humidy with an "Auto-Refresh" of Browser
 *    Every 5 seconds.
 * 7) Contains a Built in WiFi Manager to connect Wio Device
 *    to your home router via a Graphical User Interface.
 */
// Libraries
// ---------
#include <rpcWiFi.h>
#include <DNSServer.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <WiFiManager.h>
#include"TFT_eSPI.h"
#include "DHT.h" // Groove DHT Temperature $ Humidity library
#include"seeed_line_chart.h" // Line Chart Library
#include <math.h> // Include Math Library

TFT_eSPI tft;  // Create Object

// Define Line Chart Parameters
// ----------------------------
#define max_size 50 //maximum size of data doubles data[2];
doubles data; //Initilising a doubles type to store data
TFT_eSprite spr = TFT_eSprite(&tft);  // Sprite

// Screen Colors (Reference)
// -------------------------
//#define TFT_BLACK       0x0000      /*   0,   0,   0 */
//#define TFT_NAVY        0x000F      /*   0,   0, 128 */
//#define TFT_DARKGREEN   0x03E0      /*   0, 128,   0 */
//#define TFT_DARKCYAN    0x03EF      /*   0, 128, 128 */
//#define TFT_MAROON      0x7800      /* 128,   0,   0 */
//#define TFT_PURPLE      0x780F      /* 128,   0, 128 */
//#define TFT_OLIVE       0x7BE0      /* 128, 128,   0 */
//#define TFT_LIGHTGREY   0xC618      /* 192, 192, 192 */
//#define TFT_DARKGREY    0x7BEF      /* 128, 128, 128 */
//#define TFT_BLUE        0x001F      /*   0,   0, 255 */
//#define TFT_GREEN       0x07E0      /*   0, 255,   0 */
//#define TFT_CYAN        0x07FF      /*   0, 255, 255 */
//#define TFT_RED         0xF800      /* 255,   0,   0 */
//#define TFT_MAGENTA     0xF81F      /* 255,   0, 255 */
//#define TFT_YELLOW      0xFFE0      /* 255, 255,   0 */
//#define TFT_WHITE       0xFFFF      /* 255, 255, 255 */
//#define TFT_ORANGE      0xFDA0      /* 255, 180,   0 */
//#define TFT_GREENYELLOW 0xB7E0      /* 180, 255,   0 */

// LCD Backlight
// -------------
#define LCD_BACKLIGHT (72Ul)

// Global Variables
// ----------------
float Hum; // Humidity storage Variable
```

```
float TemperatureC; // Temperature storage variable for Deg C
float TempF; // Temperature storage variable for Deg F
unsigned long startMillis;  //some global variables available anywhere in the program
unsigned long currentMillis;
const unsigned long period = 3000;  //the value is a number of milliseconds (3 seconds)
unsigned long startMillis1;  //some global variables available anywhere in the program
unsigned long currentMillis1;
const unsigned long period1 = 6000;  //the value is a number of milliseconds (6 seconds)

// DHT Sensor Characteristics (Uncomment whatever type you're using)
// -----------------------------------------------------------------
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
//#define DHTTYPE DHT10   // DHT 10
//#define DHTTYPE DHT20   // DHT 20
#define DHTPIN D0     // Data Pin we're connected to
DHT dht(DHTPIN, DHTTYPE);   //   DHT11 DHT21 DHT22
//DHT dht(DHTTYPE);         //   DHT10 DHT20 don't need to define Pin

// CallBack Routine
// ================
void configModeCallback (WiFiManager *myWiFiManager) {
  Serial.println("Entered config mode");
  Serial.println(WiFi.softAPIP());
  //if you used auto generated SSID, print it
  Serial.println(myWiFiManager->getConfigPortalSSID());
  tft.fillScreen(TFT_BLACK); // Clear Screen
  tft.setTextColor(TFT_WHITE);
  tft.drawString("On a phone, Tablet or PC", 10, 10); //prints strings from (x, y)
  tft.drawString("Goto Wifi Settings", 10, 30); //prints strings from (x, y)
  tft.setTextColor(TFT_GREEN);
  tft.drawString("Connect to Wio Terminal", 10, 70); //prints strings from (x, y)
  tft.setTextColor(TFT_CYAN);
  tft.drawString("Enter in SSID & Passwword", 10, 110); //prints strings from (x, y)
  tft.drawString("In Graphical Interface", 10, 130); //prints strings from (x, y)
  tft.setTextColor(TFT_RED);
  tft.drawString("If no Graphical Interface", 10, 170); //prints strings from (x, y)
  tft.drawString("Type 192.168.1.1", 10, 190); //prints strings from (x, y)
  tft.drawString("Inside a WEB Browser", 10, 210); //prints strings from (x, y)
  delay(1000); // 1 second delay
}

WebServer server(80);  // Create Server on Port 80

// Read Sensor Function
// --------------------
void SensorData(){
Hum = dht.readHumidity(); // Measure the humidity
Serial.println("Humidity = " + String(Hum));
TemperatureC = dht.readTemperature(); // Measure the temperature
TempF = ((TemperatureC * 9/5) + 32); // Convert temperature to degrees Fahrenheit
Serial.println("Temperature = " + String(TempF));
// Compare temperature & humidity events and perform a check sum.
if (isnan(TemperatureC) || isnan(Hum)){ // Print "0" for a bad reading
  TempF = 0;
  Hum = 0;
  tft.fillScreen(TFT_BLACK); // Clear Screen
  tft.setTextSize(1);
  tft.setTextColor(TFT_RED);
  tft.drawString("Bad Readings on Sensor", 10, 10); //prints strings from (x, y)
  tft.drawString("Check Connections", 10, 50); //prints strings from (x, y)
  tft.drawString("Check Sensor", 10, 70); //prints strings from (x, y)
  delay(2000);
  tft.fillScreen(TFT_BLACK); // Clear Screen
  }
}

// Funtion to handle Root Page
```

```
// ---------------------------
void handleRoot() {
  char temp[800];
  float TEMPERATURE = TempF;
  float HUMIDITY = Hum;
  //Serial.println(TEMPERATURE + " " + HUMIDITY);

  snprintf(temp, 800,

           "<html>\
  <head>\
    <meta http-equiv='refresh' content='5'/>\
  <style>\
      body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif;
      Color: #000088; }\
    </style>\
</head>\
<body>\
  <h2>Wio Terminal DHT WebServer</h2>\
  <p>Temperature (Deg F): %.2f</p>\
  <p>Relative Humidity : %.2f</p>\
 </body>\
</html>",
           TEMPERATURE, HUMIDITY
          );
  server.send(200, "text/html", temp);
}

void handleNotFound() {
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";

  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }

  server.send(404, "text/plain", message);
}

// Main Program
// ============
void setup() {
  Serial.begin(115200);
  digitalWrite(LCD_BACKLIGHT, LOW);
  pinMode(WIO_KEY_C, INPUT_PULLUP);  // Initialize top Left button
  tft.begin();
  tft.setRotation(3);
  spr.createSprite(TFT_HEIGHT,TFT_WIDTH);
  tft.fillScreen(TFT_BLACK); // Clear Screen
  tft.setTextSize(2);
  tft.setTextColor(TFT_WHITE);
  tft.drawString("WiFi DHT Server", 10, 10); //prints strings from (x, y)
  tft.drawString("By Roy Guerra", 10, 40); //prints strings from (x, y)
  delay(2000);  // Delay 2 seconds
  tft.fillScreen(TFT_BLACK); // Clear Screen
  WiFiManager wifiManager;
    if (digitalRead(WIO_KEY_C) == LOW) {
    Serial.println("WiFi Reset");
    wifiManager.resetSettings();
    tft.fillScreen(TFT_BLACK); // Clear Screen
    tft.setTextColor(TFT_RED);
    tft.drawString("WiFi Settings Are Reset", 10, 30); //prints strings from (x, y)
    tft.drawString("Turn Off Power Button", 10, 50); //prints strings from (x, y)
```

```
      tft.drawString("Re-Start The Wio Device", 10, 70); //prints strings from (x, y)
      }
    //delay(2000);  // Delay 2 seconds
      //set callback that gets called when connecting to previous WiFi fails, and enters Access Point mode
      wifiManager.setAPCallback(configModeCallback);
      //Fetches ssid and pass from RTL8720 and tries to connect
      //if it does not connect it starts an access point with the specified name
      //here  "AutoConnectAP"
      //and goes into a blocking loop awaiting configuration
      // delay(2000);  // Delay 2 seconds
      wifiManager.autoConnect("Wio Terminal");
      //if you get here you have connected to the WiFi
      Serial.println("WiFi Is Connected");
      Serial.println("IP Address = ");
      Serial.println(WiFi.localIP());
      Serial.println("SSID = ");
      Serial.println(WiFi.SSID());
      long rssi = WiFi.RSSI();
      Serial.println("RSSI = ");
      Serial.println(WiFi.RSSI());
      tft.fillScreen(TFT_BLACK); // Clear Screen
      tft.setTextColor(TFT_YELLOW);
      tft.drawString("Wifi Connected", 10, 30); //prints strings from (x, y)
      tft.setTextColor(TFT_CYAN);
      tft.drawString("SSID = " + String(WiFi.SSID()), 10, 70); //prints strings from (x, y)
      tft.setTextColor(TFT_MAGENTA);
      tft.drawString("IP Add = " + String(WiFi.localIP().toString()), 10, 110); //prints strings from (x, y)
      tft.setTextColor(TFT_BLUE);
      tft.drawString("RSSI = " + String(rssi) + " dBm", 10, 150); //prints strings from (x, y)
      delay(5000);  // 5 second Delay
      tft.fillScreen(TFT_BLACK); // Clear Screen
    Serial.println("");
    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
    dht.begin(); // Initialize DHT sensor
    startMillis = millis();  //initial time stamp for sensor readings
    server.on("/", handleRoot);
    server.on("/inline", []() {
      server.send(200, "text/plain", "this works as well");
    });
    server.onNotFound(handleNotFound);
    server.begin();
    Serial.println("HTTP server started");
    tft.setTextSize(2);
    tft.fillScreen(TFT_BLACK); // Clear Screen
    tft.setTextColor(TFT_GREEN);
    tft.drawString("HTTP Server Started", 10, 10); //prints strings from (x, y)
    delay(2000);
    tft.fillScreen(TFT_BLACK); // Clear Screen
  }

  void loop() {
    currentMillis = millis();  // Get a time Stamp
    currentMillis1 = millis(); // Get a time Stamp
    if (currentMillis - startMillis >= period){  // Test whether the period has elapsed
      SensorData();  // Goto Function
      if (WiFi.status() != WL_CONNECTED) {
      tft.fillScreen(TFT_BLACK); // Clear Screen
      tft.setTextSize(1);
      tft.setTextColor(TFT_RED);
```

```
273       tft.drawString("WiFI Network Has Dropped", 10, 10); //prints strings from (x, y)
274       tft.drawString("Press Reset If WiFi", 10, 50); //prints strings from (x, y)
275       tft.drawString("Doesn't Come Back In 5s", 10, 70); //prints strings from (x, y)
276       WiFi.reconnect();
277       delay(2000);
278       tft.fillScreen(TFT_BLACK); // Clear Screen
279       }
280       tft.setTextSize(2);
281       tft.setTextColor(TFT_WHITE); // TFT Color
282       tft.drawString("TEMP & Humidty Parameters", 10, 10); //prints strings from (x, y)
283       tft.drawString("========================", 10, 30); //prints strings from (x, y)
284       tft.setTextColor(TFT_MAGENTA); // TFT Color
285       //Serial.println("Temperature = " + String(TempF)); // Debug
286       tft.drawString("Temperature = " + String(TempF,1) + "_F", 10, 60); //Text to
          Display to 1 decimal
287       tft.setTextColor(TFT_CYAN); // TFT Color
288       //Serial.println("Humidity = " + String(Hum)); // Debug
289       tft.drawString("Humidity = " + String(Hum,1) + "%", 10, 90); //Text to Display to 1
          decimal
290       tft.setTextSize(1);
291       tft.setTextColor(TFT_YELLOW);
292       tft.drawString(" Note 1: On a PC, Tablet, Phone; Open A WebBrower", 10, 150);
          //prints strings from (x, y)
293       tft.drawString("         And Type In The Wio Device IP Address to", 10, 160);
          //prints strings from (x, y)
294       tft.drawString("         Display the Server Information", 10, 170); //prints
          strings from (x, y)
295       tft.setTextColor(TFT_GREEN);
296       tft.drawString(" Note 2: To Reset / Erase WiFi Settings; Press Top", 10, 200);
          //prints strings from (x, y)
297       tft.drawString("         Left Buttonn and Cycle Power to Wio Device", 10, 210);
          //prints strings from (x, y)
298
299       /* Note - Line Chart not used, conflicts with Server library
300
301       spr.fillSprite(TFT_WHITE); // Previous = TFT_DARKCYAN
302       if (data.size() == max_size) {
303           data.pop();//this is used to remove the first read variable
304       }
305       data.push(Hum); //read variables and store in data
306
307       //Settings for the line graph title
308       auto header =  text(0, 0)
309                   .value("Humidity Graph")
310                   .align(center)
311                   .color(TFT_DARKGREEN)
312                   .valign(vcenter)
313                   .width(tft.width())
314                   .thickness(1);  // Previous = 2
315       header.height(header.font_height() * 8);  // Previous = 4
316       header.draw(); //Header height is the twice the height of the font
317     //Settings for the line graph
318       auto content = line_chart(20, header.height()); //(x,y) where the line graph begins
319            content
320                   .height(tft.height() - header.height() * 1.5) //actual height of the
                      line chart
321                   .width(tft.width() - content.x() * 2) //actual width of the line chart
322                   .based_on(0.0) //Starting point of y-axis, must be a float
323                   .show_circle(true) //drawing a cirle at each point, default is on.
324                   .y_role_color(TFT_WHITE)
325                   .x_role_color(TFT_WHITE)
326                   .value(data) //passing through the data to line graph
327                   .color(TFT_RED) //Setting the color for the line
328                   .draw();
329       spr.pushSprite(0, 30); // x,y
330       delay(50); // 3 Sec delay
331      */
332      // tft.setTextSize(1);
333      // tft.setTextColor(TFT_BLACK); // TFT Color to Clear Previous Text
```

```
334       // tft.drawString("Temperature = " + String(TempF,1) + "_F", 10, 10); //Text to
          Display to 1 decimal
335       // tft.setTextColor(TFT_BLACK); // TFT Color to Clear Previous Text
336       // tft.drawString("Humidity = " + String(Hum,1) + "%", 180, 10); //Text to Display
          to 1 decimal
337        startMillis = currentMillis;  // New Time Stamp
338      }
339       if (currentMillis1 - startMillis1 >= period1){  // Test whether the period has
          elapsed
340         tft.fillScreen(TFT_BLACK); // Clear Screen
341        startMillis1 = currentMillis1;  // New Time Stamp
342       }
343      server.handleClient();
344    }
345
```